基于非线性渐变式颜色映射的 LIC 改进算法 *

高茂庭，董红周，周　凡

(上海海事大学 信息工程学院，上海 201306)

摘　要：线积分卷积（LIC）算法展现的纹理反映了整个矢量场的方向结构，但却不能展现矢量场的强度大小。针对此问题提出基于非线性渐变式颜色映射的 LIC 改进算法，将矢量场强度与白噪声结合作为 LIC 的输入纹理，运用 FastLIC 思想并划分纹理区域同步执行 LIC 运算来提高算法效率；再将矢量场强度作非线性变换，根据渐变式颜色映射方案使用 OpenCV 处理引擎并行实现矢量场强度的颜色映射； 最后由 LIC 得到的灰度纹理和颜色映射结果确定合成系数并构造累计函数增强二者结果，再进行线性合成运算得到最终的可视化效果。在对全球海洋流场和风场两种典型的矢量场进行可视化以及与其他算法对比实验表明，改进算法得到的可视化效果纹理清晰，较好地展示出矢量场的方向和强度，能够更准确地反映矢量场的全方位信息和局部变化情况。

关键词：线积分卷积；矢量场；场强；可视化；累计函数

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.03.0199

Improved LIC algorithm based on nonlinear gradual-changing color mapping

Gao Maoting, Dong Hongzhou, Zhou Fan

(*College of Information Engineering, Shanghai Maritime University, Shanghai* 201306, *China*)

Abstract: The texture revealed by line integral convolution (LIC) reflects the directional structure of the whole vector field, but cannot show the field intensity. In order to solve this problem, this paper proposed an improved LIC algorithm based on nonlinear gradual-changing color mapping. Firstly, the intensity of vector field combined with white noise to form the input texture of improved LIC, which used the idea of FastLIC and divides the texture into several regions for synchronous execution of LIC, to improve the efficiency of LIC. Secondly, the improved LIC does a non-linear transformation on the intensity of vector field, and used the processing engine, OpenCV, to implement the color mapping of the vector field intensity according to the gradual-changing color-mapping scheme. Finally, the results of gray texture obtained by LIC and color mapping determine the synthetic coefficient, constructing a cumulative function enhances the two results, and adopting the way of linear synthesis obtains the final visualization. Simulation results show that when applied to the global ocean flow field and wind field, the proposed algorithm generates a clear visualization displays the direction and intensity of vector field better, and reflects the global information and local changes of the vector field better, when compared with other algorithms.

Key words: line integral convolution; vector field; intensity; visualization; cumulative function

0　引言

采用可视化的方式对矢量场数据展示往往是掌握和分析其运动规律的重要手段，也为进一步发现矢量场的特征结构提供了基础支撑。但矢量数据因具有大小和方向两个属性，目前还没有直接适用的可视化模型[1]。实际应用中的矢量场(如海洋流场、风场等)往往具有复杂性和特殊性，传统的点图标法和矢量线法[2]已经无法满足其可视化要求。而线积分卷积算法（line integral convolution, LIC）[3]通过生成连续平滑的纹理展现矢量场的方向变化和形状特征，在矢量场可视化中具有独特的优势[4]。

对矢量场数据可视化需要同时考虑方向和强度大小，但 LIC 算法生成的纹理只能展示矢量场的方向，并不能反映出矢量场的强度信息，而且算法比较耗时。针对 LIC 算法存在的问题，许多改进算法先后被提出，张文等人[5]提出了基于 HLS 模型的 FastLIC 算法，将矢量大小通过 HLS 颜色模型映射，再利用 OpenGL 技术将映射结果叠加到 FastLIC 算法生成的灰度纹理上，为 LIC 算法展现矢量场强度提供了新思路；陆剑锋等人

收稿日期：2018-03-21；**修回日期：**2018-05-04　　**基金项目：**国家自然科学基金资助项目（41701523）

作者简介：高茂庭（1963-），男，江西九江人，教授，博士，主要研究方向为智能信息处理、数据库、信息系统（mtgao@163.com）；董红周（1992-），男，江苏连云港人，硕士研究生，主要研究方向为数据可视化、数据库、信息系统；周凡（1986-），男，湖北广水人，讲师，博士，主要研究方向为摄影测量、计算机视觉.

[6]提出了基于对比度数量映射的 LIC 算法，通过图像的对比度对矢量场强度进行数量映射，再与 LIC 纹理合成，结果同时展示了矢量场的方向和大小；张文耀等人[7]提出了基于 HSV 模型的 LIC 算法，将矢量场方向映射为色度，场强大小映射为饱和度，先得到矢量场中各点的 HSV 颜色值，再转换为相应的 RGB 值，得到最终的颜色纹理，并通过 FastLIC 算法提高效率；赵安元等人[8]提出了基于查找表的 LIC 算法，通过建立查找表提前存储流线的方式提高算法效率，将矢量场强度映射为颜色值，与 LIC 得到的灰度值加权平均得到最终的纹理值；詹芳芳等人[9]提出了基于 CUDA 架构的颜色增强 LIC 算法，通过在 CUDA 架构上并行实现 LIC 算法来提高算法的效率，用分类颜色映射方法对矢量场强度进行映射，不仅展示了矢量场的方向和强度，还突出了用户感兴趣区域的矢量场特征；Ding 等人[10]提出了基于非稳定流场的并行 LIC 算法，通过设计流线重用策略和建立值散射收集机制提高 LIC 算法的效率，再利用 GPU 的并行性对流场强度进行线性映射，实现了高密集流场的可视化；韩敏等人[11]提出了基于 GPU 的流线增强型 LIC 算法，根据积分点所在区域结合不同的积分方法，自适应更新积分步长来减小 LIC 的计算量，利用 GPU 的并行性提高渲染速度，得到的可视化效果可以直观地反映出矢量场的特征分布。

上述 LIC 改进算法通过将矢量场强度进行颜色映射，再与 LIC 得到的灰度纹理线性合成得到最终的可视化效果，虽然同时展示了矢量场的方向和强度，但灰度纹理与色彩结合后亮度会变小，导致整体颜色变暗，对比度、视觉分辨度降低；当局部区域矢量场强度变化平缓时，表示场强大小的颜色将难以区分；线性合成系数也没有合适的确定方法，如果灰度纹理的系数过高，则颜色不明显，反之，纹理将不清晰。

因此，本文提出一种基于非线性渐变式颜色映射的 LIC 改进算法，将矢量场强度与白噪声结合形成新的输入纹理，结合 FastLIC 算法[12]思想并划分纹理区域同步执行 LIC 运算；通过 OpenCV 处理引擎并行实现矢量场强度的非线性渐变式颜色映射；再根据 LIC 得到的灰度纹理值和非线性渐变式颜色映射结果确定线性合成系数，并构造附加累计函数将两者结果分别根据函数进行变换，最后合成运算得到最终的彩色纹理可视化效果。

1　经典线积分卷积算法

线积分卷积算法（line integral convolution,LIC）是 Briam Cabral 和 Leith Leedom 在 SIGGRAPH93 上提出的，这种用卷积形式表示矢量场的方向信息来源于一种运动模糊的思想。LIC 算法作为一种基于纹理的矢量场可视化方法，综合了几何形状映射和颜色映射的长处，生成的纹理在空间上连续平滑，能够清晰地表现出矢量场的方向变化和形状特征，克服了传统点图标法、几何矢量线法等可视化效果混乱的缺点[13]。

在 LIC 运算中，首先生成一个与矢量场分辨率相同的白噪声作为输入纹理，然后在矢量场中选取点生成流线，即将该点沿矢量场正、反方向对称伸展得到局部流线，将流线上所有对应的输入噪声值按卷积核参与卷积，卷积结果作为输出纹理的灰度值。假定生成的局部流线用 $\sigma(s)$ 表示，s 是弧长参数，T 为输入的白噪声纹理，LIC 纹理在 $x_0=\sigma(s)$ 处的灰度值为 $I(x_0)$，则 $I(x_0)$ 计算如式(1)所示。

$$I(x_0)=\frac{1}{Z}\int_{s_0-L}^{s_0+L}k(s-s_0)T(\sigma(s))ds \quad (1)$$

其中：$Z=\int_{-L}^{L}k(s)ds$ 为归一化参数；$k(s)$，$s\in[-L,L]$ 为卷积核函数，不同的卷积核函数将对应不同的卷积结果[14]。经典 LIC 算法的流程如图 1 所示。

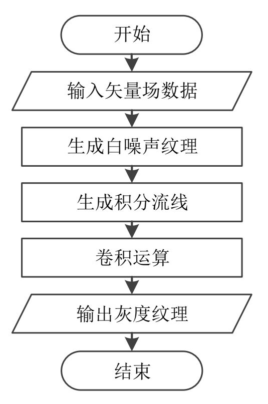


图 1　经典 LIC 算法的流程

2　基于非线性渐变式颜色映射的 LIC 改进算法

为了更好地显示矢量场的方向和强度信息，获得高分辨率的可视化效果，同时提高 LIC 算法的效率，本文提出一种基于非线性渐变式颜色映射的 LIC 改进算法。该算法分为三个模块，如图 2 所示。

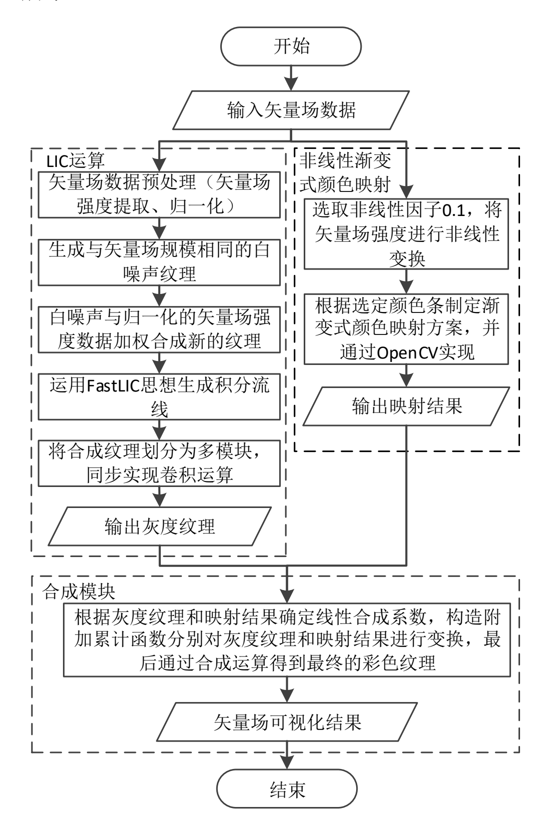


图 2　基于非线性渐变式颜色映射的 LIC 改进算法流程

第一个模块是 LIC 运算的改进，将矢量场强度与白噪声相结合形成新的纹理作为 LIC 运算的输入，在 LIC 得到的灰度纹理中融入场强信息进行增强显示。为了提高 LIC 的运算速度，降低算法对硬件的依赖程度，该模块在 FastLIC 算法思想的基础上，将合成纹理分成若干区域同步执行 LIC 运算。

第二个模块是非线性渐变式颜色映射，根据实际应用场景选定颜色表，制定非线性渐变式颜色映射方案并采用 OpenCV 实现，克服了线性映射在矢量场强度变化不明显区域映射效果不佳的缺点。该模块与第一模块将实现并行计算来提高算法的运行速度。

第三个模块是灰度纹理与非线性渐变式颜色映射结果合成，首先根据各个点的灰度纹理值和映射结果确定线性合成系数；再通过构造附加的累计函数，将灰度纹理值和映射结果根据函数作相应变换；最后合成运算得到最终的彩色纹理结果，克服了一般线性合成方法所得效果亮度暗、纹理不清晰以及合成系数难以确定等缺点。

2.1 LIC 运算的改进

经典 LIC 算法获取原始矢量场后，生成一个与原始矢量场规模一致的白噪声作为输入纹理。假设二维矢量场大小为 $X \times Y$，由原始矢量场得到的白噪声纹理为 T，为了计算方便，用 $rand(x,y)$ 表示 T 中各个项的取值，$rand(x,y)$ 为介于 $[0,1]$ 之间的随机数，则可令 T 为

$$\boldsymbol{T}=\begin{bmatrix} rand(1,1) & rand(1,2) & \cdots & rand(1,y) \\ rand(2,1) & rand(2,2) & \cdots & rand(2,y) \\ \vdots & \vdots & \ddots & \vdots \\ rand(x,1) & rand(x,2) & \cdots & rand(x,y) \end{bmatrix}$$

文献[15]提出将白噪声纹理着色来区分不同的运动方向，本文方法在白噪声纹理中融入原始矢量场的强度信息进行 LIC 运算。由式(1)可得在白噪声纹理中融入场强大小后形成新的输入纹理完全可行。通过原始矢量场获得场强矩阵后，为了能够在白噪声矩阵 T 中融入场强大小，需要对原始矢量场强度矩阵进行归一化处理[16]。定义原始矢量场强度矩阵为 M，归一化处理后的矩阵为 N，现给出将矩阵 M 中所有项归一化 $[a,b]$ 区间的一般方法：

$$n_{xy}=a+\frac{(b-a)}{m_{\max}-m_{\min}}(m_{xy}-m_{\min}) \tag{2}$$

其中：m_{xy} 为矩阵 M 中的任意项；n_{xy} 为 m_{xy} 归一化后对应的项，$n_{xy}\in[a,b]$；$m_{\max}$ 为矩阵 M 中项的最大值；$m_{\min}$ 为矩阵 M 中项的最小值。最后所得矩阵 N 为

$$\boldsymbol{N}=\begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1y} \\ n_{21} & n_{22} & \cdots & n_{2y} \\ \vdots & \vdots & \ddots & \vdots \\ n_{x1} & n_{x2} & \cdots & n_{xy} \end{bmatrix}$$

由于矩阵 T 中的 $rand(x,y)\in[0,1]$，为了白噪声与矢量场强度更好地融合，此处应该把场强矩阵 N 中的值归一化到 $[0,1]$ 内，即令式(2)中 $a=0, b=1$。由式(1)得 LIC 算法输出的纹理强度 $I(x_0)$ 与白噪声 T 成正相关，在融入场强大小后，$I(x_0)$ 和新生成的纹理依然要保证这种相关性。所以本文采用将白噪声矩阵 T 与归一化的矢量场强度矩阵 N 进行加权求和的方式，矩阵 T 中的各项取值为 $[0,1]$，M 归一化后各项数值的区间为 $[0,1]$。根据矩阵 T，N 定义一个生成新的输入纹理的函数 $g(T,N)$，令权重系数为 α,β，白噪声融入场强大小后形成的新的纹理为 Q，则

$$Q=g(T,N)=\frac{\alpha T+\beta N}{\alpha+\beta} \tag{3}$$

α 越大，表示白噪声所占比重越高，由此生成的输出纹理将越能体现矢量场的方向信息；β 越大，表示矢量场强度所占比重越高，最终生成的可视化效果越能体现出矢量场的强度信息。

获得新的输入纹理后，LIC 算法需要从当前点开始分别向正、反两个方向延伸到两段长为 L 的流线，此过程称为流线生成[17]。本文选用数值积分的方式生成流线。在对海洋流场和风场可视化的应用中，由于实测数据是基于采样点的，这里需要用离散的思想描述 LIC 算法中流线的轨迹 $s(t)$，用 $V(s(t),t)$ 表示流线速度，则在一个时间段 $[t_k,t_{k+1}]$ 内，流线的轨迹可表示为

$$s(t_{k+1})=s(t_k)+\int_{t_k}^{t_{k+1}}V(s(t),t)dt \tag{4}$$

其中：$V(s(t),t)$ 又可以分解为

$$V(s(t),t)=[v_x([x,y],t),v_y([x,y],t)] \tag{5}$$

对式(4)进行求解即可得到该点生成的流线。为了对矢量场中复杂微小结构进行很好地展示，本文选用精度最高的变长四阶 Runge-Kutta 方法[18]求出生成流线，而且随着矢量场数据规模的增加，基于 RK4 的求解方法加速效果明显。

然而流线生成模块因重复计算大量流线导致 LIC 算法十分耗时。为了提高算法的运行效率，本文对 LIC 运算进行了改进。目前降低 LIC 算法时间复杂度的方式主要分为四类：第一类是将 LIC 算法改为 FastLIC 算法，不用重复计算流线，而是充分利用了点之间的相关性，即后一点的纹理值可以由前一个点得出；第二类是并行 LIC 算法，在大规模并行分布式存储计算机上实现，速度快但对硬件要求特别高；第三类是分级线积分策略[19]，利用多核架构对算法中的积分过程进行加速，缩短 LIC 运算时间；第四类则是通过 GPU 实现 LIC 算法的并行计算，利用 GPU 大量的执行单元提高算法效率。第二类和第四类对硬件要求过高，对于第三类改进算法，如果矢量场规模增加，则运算时间依旧呈线性增长。本文选择 FastLIC 算法思想来提高 LIC 算法的效率，并在此基础上将输入纹理划分为 n 个区域，分模块同步执行 LIC 运算。FastLIC 算法思想提出，根据相邻位置点之间的相关性，可以通过已知点的灰度值求其相邻点的灰度值，避免不必要的流线生成计算。假设 x_1 和 x_2 是矢量场上

相邻的两点，计算它们的卷积时，可以发现这两点生成的流线有很大部分相互覆盖的区域，如图 3 所示。

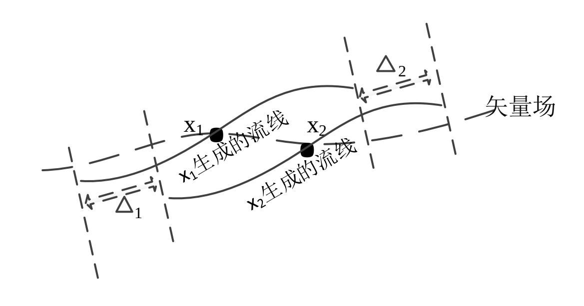


图 3　点和 x_2 点生成的流线

若已知 x_1 处的灰度值为 $I(x_1)$，则 x_2 处的灰度值 $I(x_2)$ 可通过式(6)求得

$$I(x_2)=I(x_1)-I(\Delta_1)+I(\Delta_2) \tag{6}$$

式(6)只在盒式卷积核的情况下成立，且 $I(\Delta_1)$ 和 $I(\Delta_2)$ 分别是 Δ_1 和 Δ_2 区域的卷积结果。由于数据采样点具有离散的特性，通过式(1)可得到采样点离散情况下 x_1 点处纹理灰度值 $I(x_1)$ 的计算方法如式(7)所示。

$$I(x_1)=\sum_{i=-L}^{L}T(x_i)k(x_i) \tag{7}$$

其中：x_i 为流线上离散点；$T(x_i)$ 为 x_i 点对应的输入纹理；$k(x_i)$ 为 x_i 对 x_1 的贡献且满足 $\int_{-L}^{L}k(x_i)=1$。

根据采样点离散的特性，若每条长为 L 的流线上有 n 个点，则正反流线上共有 $2n+1$ 个点。对于离散点的卷积，需要满足所有参与卷积的点的卷积系数之和为 1，且每个点的系数都相等，所以得到系数为 $1/(2n+1)$，即式(7)中的 $k(x_i)=1/(2n+1)$，满足盒式卷积核，不仅保证了显示质量同时得到最快的运算速度，所以 x_1 点处的灰度值为

$$I(x_1)=\frac{1}{2n+1}(\sum_{i=-n}^{n}T(x_i)) \tag{8}$$

将式(3)代入，得到 x_1 点处的灰度值计算为

$$I(x_1)=\frac{1}{2n+1}(\sum_{i=-n}^{n}\frac{\alpha T(x_i)+\beta N(x_i)}{\alpha+\beta}) \tag{9}$$

所以，运用 FastLIC 思想，基于盒式卷积核，对于任意从点 x_m 开始的 LIC 运算，不需要重复计算大量流线，只要求出 $I(x_m)$ 的值，$I(x_{m+1})$ 和 $I(x_{m-1})$ 的值便可通过式(10)(11)求得，且求解采用嵌入式变步长龙格塔库数值积分法，提高了流线计算的准确性。

$$I(x_{m+1})=I(x_m)+k(Q(x_{m+1+L})-Q(x_{m-L})) \tag{10}$$

$$I(x_{m-1})=I(x_m)+k(Q(x_{m-1-L})-Q(x_{m+L})) \tag{11}$$

式(10)(11)中：$k=1/(2n+1)$；$Q(x_i)$ 为式(3)计算得到的新纹理。

2.2　非线性渐变式颜色映射

为了更好地展现矢量场的强度信息，使视觉效果更佳，本文通过自选定颜色表制定非线性渐变式颜色映射方案对矢量场强度进行颜色渲染并采用 OpenCV 实现。OpenCV 是一个跨平台的计算机视觉库，其强大的视觉处理算法能够准确地从选定的颜色表中获取颜色样本，不需要添加新的外部支持也可以完整地编译链接生成执行程序，在颜色映射方面处理十分高效。本文将借助 OpenCV 轻量级处理引擎快速实现非线性渐变式颜色映射方案。为了进一步提高算法的效率，颜色映射过程与 LIC 运算采用并行的处理方式。

2.2.1 非线性场强值转换

在实际应用中，矢量场强度往往分布不均匀，如果场强集中在一段较小的范围内，线性映射会将场强映射结果集中在某一种颜色区域内，这样就很难区分出局部场强的大小。针对线性映射存在的缺点，本文通过非线性方式转换场强值。从原始矢量场中得到场强大小 mag，将场强大小根据式(12)进行转换：

$$newmag=\frac{c^{mag/mag_{max}}-1}{c-1} \tag{12}$$

其中：$newmag$ 为非线性变换后的结果；c 为非线性映射因子，实验一般取 $c=0.1$；mag_{max} 为矢量场强度中的最大值。根据 $c=0.1$ 得到式(12)图像如图 4 所示，由图像可得 $newmag\in[0,1]$。

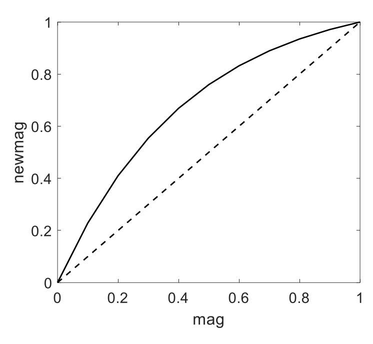


图 4　c=0.1 时，newmag 与 mag 的映射关系

2.2.2 渐变式颜色映射

由于本文的实验对象是海洋流场和风场，为了得到较好的颜色映射效果，本文的颜色模型将参照 PanoplyWin[20]软件中提供的颜色表。PanoplyWin 软件是由 NASA.GISS 发布的海洋数据可视化软件，拥有专业的海洋要素颜色映射体系。从 PanoplyWin 软件中选取对应海洋流场和风场映射的颜色表如图 5 所示。

图 5　PanoplyWin 软件中对应流场和风场映射的颜色表

在原始场强经过非线性变换得到新的场强值后，为了实现颜色映射的平滑过渡，本文提出如下的渐变式颜色映射方案：

由 OpenCV 读取出颜色模型的大小 $w\times h$，若要将非线性变换后的 $newmag$ 根据颜色表进行映射，还需将 $newmag$ 按照式(13)作放大处理：

$$p=\lfloor newmag\times w\rfloor \tag{13}$$

p 为 $newmag$ 变换后的值，$p\in[0,w]$。将颜色表的长、宽分别等分成 w、h 份，取其长度上任意相邻的两点，记为 p_m, p_{m+1}，假设 p 落在 p_m, p_{m+1} 之间，p 到 p_m 的距离为 l_1，p

到 p_{m+1} 的距离为 l_2，如图 6 所示。

p_m —— l_1 —— p —— l_2 —— P_{m+1}

图 6 在颜色表的长度上任取两个相邻的点

若要达到渐变效果，则 p 处的颜色取值可由式(14)(15)得出

$$color[p]=\frac{l_2}{l_1+l_2}*color[p_m]+\frac{l_1}{l_1+l_2}*color[p_{m+1}] \tag{14}$$

$$color[p_m]=\frac{1}{h}\sum_{i=0}^{h-1}color[p_m,i] \tag{15}$$

式(14)(15)中：$color[p_m,i]$ 为颜色表在 (p_m,i) 处的颜色值。将 l_1 和 l_2 的取值代入式(14)得到原始矢量场中强度 mag 对应的颜色映射值为

$$\begin{aligned}color[p]=&(p_{m+1}-newmag*w)*color[p_m]\\&+(newmag*w-p_m)color[p_{m+1}]\end{aligned} \tag{16}$$

2.3 灰度纹理与非线性渐变式颜色映射结果合成

在两个模块分别执行完后，需要进行合成才能得到最终的可视化效果。令 LIC 得到的灰度纹理为 R_{LIC}，颜色映射结果为 R_{color}，常见的合成方式如式(17)(18)所示。

$$R_{last}=tR_{LIC}+(1-t)R_{color} \tag{17}$$

$$R_{last}=t\times R_{LIC}\times R_{color} \tag{18}$$

式(17)(18)中：R_{last} 为合成的最终结果；t 为合成系数，且 $t\in(0,1)$。

通过式(17)的合成，虽然可以通过参数 t 来控制灰度纹理和颜色映射结果的显示比重，但得到的效果整体亮度偏暗，对比度、视觉分辨度降低，局部纹理和颜色不清晰等。通过式(18)的合成，不仅同时弱化了纹理和颜色映射的效果，而且运算速度大大降低。由此可知，上述两种方式的合成效果并不理想。

本文通过构造附加累计分布函数来增强灰度纹理与颜色映射结果的显示。过程如下：

a) 已知 R_{LIC} 的大小为 $X*Y$，得到 R_{LIC} 中的最大灰度值为 $I_{LIC\max}$，最小灰度值为 $I_{LIC\min}$。

b) 将 R_{LIC} 中的灰度划分为 n 个等份，并标号为 $L_1,L_2,L_3,\cdots,L_n$，每个等份的区间长度为 $(I_{LIC\max}-I_{LIC\min})/n$。

c) 通过统计获得 R_{LIC} 中灰度值落在 $L_1,L_2,L_3,\cdots,L_n$ 中点的个数，记为 $h(L_i),i=1,2,3,\cdots,n$，并分别求出 $h(L_i)$ 占总数的百分比，记为 $hs(L_i)=h(L_i)/(X*Y)$，$i=1,2,3,\cdots,n$。

d) 计算各个灰度级的累计分布 $hp(L_i)$，即 $hp(L_i)=\sum_{j=0}^{i}hs(L_j)$，$i=1,2,3,\cdots,n$。

e) 重新分配灰度值 $I'(x_0)$，因为灰度值的取值为 $[0,255]$，所以实验时 n 取 255，即

$$I'(x_0)=\begin{cases}255*\sum_{j=0}^{i}\frac{h(L_j)}{x*y},i=1,2,3,\cdots,n,L_j\neq 0.\\0,\ L_j=0\end{cases} \tag{19}$$

假设 R_{LIC} 中有两个像素点 R_m,R_n，且 $I(R_m)<I(R_n)$，因为 $hp(L_i)$，$i=1,2,3,\cdots,n$ 在定义域内为增函数，将 R_m,R_n 的值带入上述式(19)中，得 $I'(R_m)<I'(R_n)$，所以通过附加累计函数增强显示的过程并不改变原来灰度纹理的变化趋势。R_{color} 中处理方式同上述 R_{LIC} 的方法，结果记为 $I_c'(x_0)$，则最终的合成方式为

$$I'_{last}(x_0)=\frac{I(x_0)}{I(x_0)+I_c(x_0)}I'(x_0)+\frac{I_c(x_0)}{I(x_0)+I_c(x_0)}I_c'(x_0) \tag{20}$$

其中：$I_c(x_0)$ 为非线性渐变式颜色映射后的结果值。

3 实验结果及其分析

3.1 数据集和实验环境

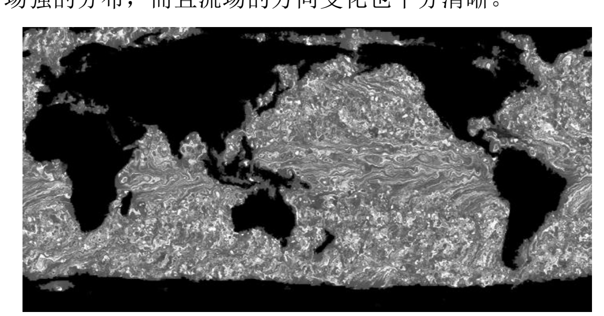
本文选取全球海洋流场和风场这两个典型的矢量场作为实验对象，采用由美国国家海洋和大气管理局(NOAA)提供的全球海洋流场和风场实测数据。数据格式为 NetCDF，时间分辨率为 6 h，空间分辨率为 $0.25^\circ\times0.25^\circ$，空间覆盖范围为 180° W~180° E，90° S~90° N。

本文的实验环境为 Windows 10 操作系统，Intel core i5 处理器和 8 GB 内存，开发工具为 VS2013，图形库选用 OpenCV，代码使用 C++语言实现。

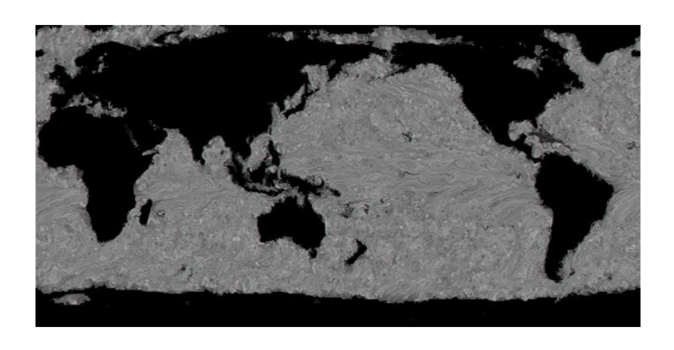
3.2 可视化效果对比实验

本文选用适合海洋流场、风场的专业颜色映射表得到的可视化效果与基于 HLS、HSV 等模型映射的 LIC 算法得到的可视化效果存在较大的颜色差，区分度很大。所以本实验是在同一颜色映射表下进行，将本文算法三个模块得到的可视化结果分别同对应的经典 LIC 算法、线性颜色映射方法以及基于线性颜色映射的 LIC 算法(LCLIC)所得结果进行比较，以此来验证本文算法在矢量场可视化应用中的优劣性。选用全球海洋流场的可视化过程进行实验说明，最后再给出本文算法应用到全球海洋风场中得到的可视化效果。实验采用控制可变因素的方式进行，经过前期多次实验比较，当划分模块的个数 n=5 时，能够较好地平衡处理速率和可视化效果质量，积分半长 L=10，$\alpha=\beta=1/2$ 时，可视化效果最佳。所以，对比实验将以此为基准进行。

在对全球海洋流场数据可视化的应用中，本文改进的 LIC 运算模块输出的结果如图 7(a)所示，经典 LIC 算法输出的结果如图 7(b)所示，黑色阴影部分为陆地面积。本文改进的 LIC 运算模块得到的可视化效果，已经通过明暗渐变的纹理初步显示了场强的分布，而且流场的方向变化也十分清晰。



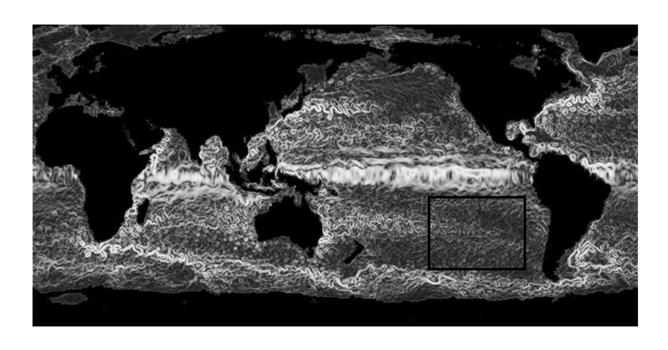
(a) 本文改进的 LIC 运算模块得到的灰度纹理效果



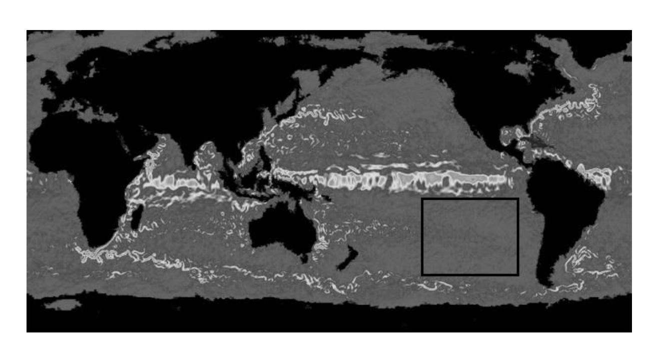
（b）经典 LIC 算法得到的灰度纹理效果

图 7　本文改进的 LIC 运算模块与经典 LIC 算法的灰度纹理对比

图 8(a)为本文非线性渐变式颜色映射的效果；(b)为采用线性方式颜色映射的效果。本文提出的非线性渐变式颜色映射方法能够适应局部流场强度变化不明显的区域。当局部区域流场强度变化缓慢时，普通的线性映射方法会在该区域形成较大一片颜色相同的区域，如图 8(b)中的方框标记区域；而基于本文的方法则可以反映出流场强度变化较小的分布情况，如图 8(a)方框标记区域。



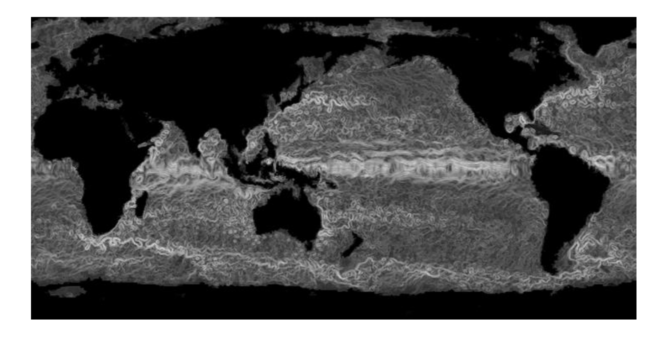
（a）非线性渐变式颜色映射



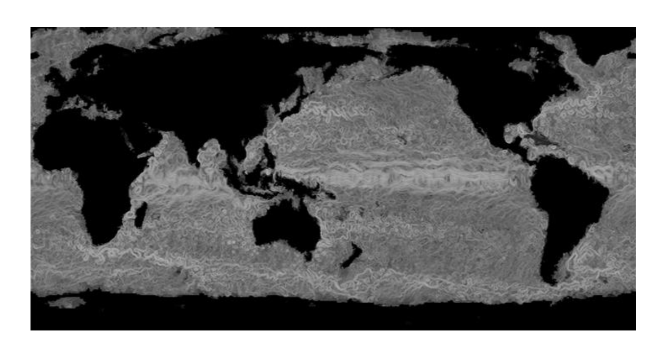
（b）普通的线性映射

图 8　非线性渐变式颜色映射与普通线性映射效果对比

LCLIC 算法是采用式(17)的合成方式，在这种合成方式中，合成系数并没有明确的规定方法。图 9(a)为式(17)中 $t=0.3$ 时 LCLIC 算法得到的可视化结果，纹理不清晰，但色彩比较丰富；图 9(b)为式(17)中 $t=0.7$ 时 LCLIC 算法得到的可视化结果，纹理效果比颜色效果明显。除了合成系数影响外，基于 LCLIC 算法得到的可视化效果仍然会整体变暗，纹理和颜色在合成后效果都会被减弱。本文算法明确了合成系数，并构造附加累计函数对两者结果进行变换，克服了合成系数难以确定以及合成效果不理想等缺点。图 10 为本文算法得到的全球流场可视化效果。



（a） $t=0.3$ 时 LCLIC 得到的全球流场可视化效果



（b） $t=0.7$ 时 LCLIC 得到的全球流场可视化效果

图 9　线性合成不同合成系数对应的可视化效果

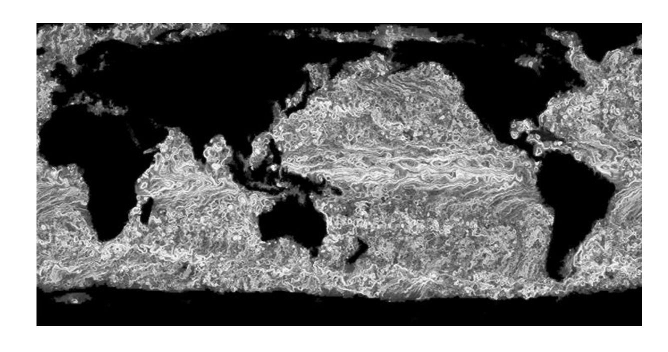


图 10　本文算法得到的全球流场可视化效果

海洋风场可视化的处理流程同海洋流场。图 11 为本文算法得到的全球海洋风场可视化效果。通过海洋流场和风场这两个复杂矢量场的可视化效果来看，本文提出的基于非线性渐变式颜色映射的 LIC 改进算法，运用在矢量场可视化中，不仅清晰地展现了矢量场的方向变化，还能表现出矢量场的强度信息。可视化效果具有较高的亮度和对比度，克服了其他 LIC 改进算法得到的彩色纹理饱和度下降、色彩偏暗、纹理区分度较低等缺点，在矢量场局部区域的可视化中，也能够清晰地展现出方向和强度的变化趋势。

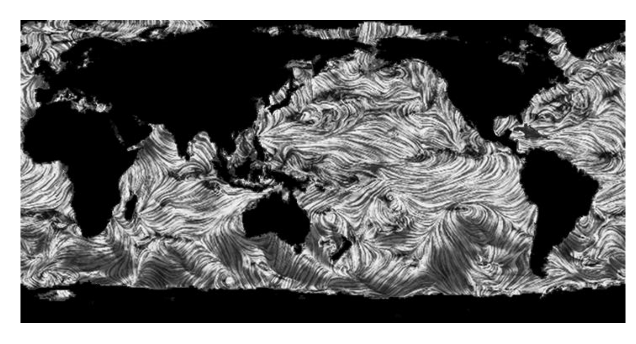


图 11　本文算法得到的全球风场可视化效果

3.3 算法效率对比实验

3.3.1 与其他可视化算法的对比实验

为了获取本文算法与其他可视化算法的时耗差距，实验选取了文献[1,2]中所提的点图标法、矢量线法以及拓扑法(分别简写为 PIM、VLM、TM)与本文算法(NGCLIC)进行对比实验。表 1 为上述可视化方法的用时统计。

表 1　不同可视化方法的用时比较/s

矢量场	PIM	VLM	TM	NGCLIC
720×361	1.02	1.36	2.41	2.03
1440×481	2.54	3.15	5.32	4.39

从时间上可以看出，本文算法的时耗比点图标法和矢量线法高，比拓扑法要低，但在本文算法得到的可视化效果中，纹理方向上的点之间具有更强的相关性，这种特征不仅体现了矢量场数据的连续性，而且达到纹理模拟真实场景的逼真效果，这是点图标法和矢量线法无法达到的。

3.3.2 与其他 LIC 改进算法的对比实验

为了分析本文算法与其他 LIC 改进算法在性能上的差异，实验又选取了文献[5～9]中所提到的 LIC 改进算法(分别简写为 HFLIC、HSVLIC、LTLIC、NCLIC)与本文算法进行对比实验。表 2 为不同的 LIC 改进算法用时统计。

表 2 不同的 LIC 改进算法用时比较/s

矢量场	HFLIC	HSVLIC	LTLIC	NCLIC	NGCLIC
720×361	2.41	2.23	2.14	1.75	2.03
1440×481	5.32	4.94	4.68	4.27	4.39

从表 2 可以看出，本文算法在保证可视化的质量的同时，比文献[5,7,8]中提到的算法在效率上有一定提升，但比文献[9]中的算法效率要低。本文算法通过运用 FastLIC 思想实现 LIC 运算模块、划分输入纹理为 n 个模块同步执行 LIC 运算、选用轻量级的 OpenCV 处理引擎、并行实现 LIC 运算和非线性渐变式颜色映射等一系列措施来提高算法的运行速度，这种优势也会随着数据量的增加而慢慢显现；但本文的 LIC 模块需要生成新的纹理，分模块同步计算需要系统开销，且算法的合成模块需要同时等到 LIC 结果和颜色映射结果才能执行，所以本文算法依旧存在时耗问题。此外，本文算法没有结合 GPU 方式进行加速渲染，这也是本文算法比文献[9]中提到的算法效率低的原因，但本文算法对硬件要求不高，适用性更广。

4　结束语

本文针对经典 LIC 算法和 LIC 改进算法在矢量场强度上映射弱的缺点，提出了基于非线性渐变式颜色映射的 LIC 改进算法。通过实验验证，改进后的算法适合变化剧烈的海洋流场、风场等典型的矢量场，不仅同时展现了矢量场的方向和强度，而且能够清晰、准确地反映出矢量场全方位信息以及局部区域的变化趋势。所绘制出的彩色纹理连续平滑，保持了较高的色彩亮度和视觉分辨率，取得了较好的可视化效果。用可视化手段处理及表现矢量环境已成为了迫切需要解决的课题。本文算法在海洋流场、风场可视化中的应用，也为气动流场、磁极场等其他矢量场提供新的研究方法。未来本文还将在此基础上充分结合数据挖掘的方法去探寻矢量场的变化规律和特征结构以及其他现象，并建立良好可视化挖掘机制。

参考文献：

[1] 王盛波，潘志庚. 二维流场可视化方法对比分析及综述 [J]. 系统仿真学报, 2014, 26 (9): 1875-1881. (Wang Shengbo, Pan Zhigeng. Comparison and analysis of visualization methods for two-dimensional flow fields [J]. Journal of System Simulation, 2014, 26 (9): 1875-1881.)

[2] 陈为，张嵩，鲁爱东. 数据可视化的基本原理与方法 [M]. 北京：科学出版社, 2013: 100-136. (Chen Wei, Zhang Song, Lu Aidong. Basic principles and methods of data visualization [M]. BeiJing: Science Press, 2013: 100-136)

[3] Cabral B, Leedom C. Imaging vector fields using line integral convolution [J]. Computer Graphics, 1993, 27: 263-270.

[4] Lu D, Zhu D, Wang Z, *et al.* Efficient level of detail for texture‐based flow visualization [J]. Computer Animation & Virtual Worlds, 2016, 27 (2): 123-140.

[5] 张文，李晓梅. LIC 纹理中可视矢量大小的方法 [J]. 计算机应用, 2000 (S1): 15-17. (Zhang Wen, Li Xiaomei. Method of visual vector size in LIC textures [J]. Journal of Computer Applications, 2000 (s1): 19-21.)

[6] 陆剑锋，潘志庚，张明敏，等. 基于图像对比度数量映射的矢量场可视化算法 [J]. 系统仿真学报, 2004, 16 (7): 1502-1505. (Lu Jianfeng, Pan Zhigeng, Zhang Mingmin, *et al.* Vector field visualization algorithm based on image contrast number mapping [J]. Journal of System Simulation, 2004, 16 (7): 1502-1505.)

[7] 张文耀，蒋凌霜. 基于 HSV 颜色模型的二维流场可视化 [J]. 北京理工大学学报, 2010, 30 (3): 302-306. (Zhang Wenyao, Jiang Lingshuang. 2D flow field visualization based on HSV color model [J]. Transactions of Beijing Institute of Technology, 2010, 30 (3): 302-306.)

[8] 赵安元，任杰，刘东权. 二维和三维矢量场的可视化 [J]. 计算机应用研究, 2011, 28 (4): 1592-1597. (Zhao Anyuan, Ren Jie, Liu Dongquan. Visualization of 2D and 3D vector field [J]. Application Research of Computers, 2011, 28 (4): 1592-1597.)

[9] 詹芳芳，胡伟，袁国栋. 二维 LIC 矢量场可视化算法的研究及改进 [J]. 计算机科学, 2013, 40 (9): 257-261. (Zhan Fangfang, Hu Wei, Yuan Guodong. Improvement of 2D LIC algorithm for vector field visualization [J]. Computer Science, 2013, 40 (9): 257-261.)

[10] Ding Zi'ang, Liu Zhanping, Yu Yang, *et al.* Parallel unsteady flow line integral convolution for high-performance dense visualization [C]// Proc of IEEE Visualization Symposium. 2015: 25-30.

[11] 韩敏，张海超，边茂松，等. 流线增强型线性积分卷积流场可视化 [J]. 系统仿真学报, 2016, 28 (12): 2933-2938. (Han Min, Zhang Haichao, Bian Maosong, *et al.* Flow visualization based on enhanced streamline line integral convolution [J]. Journal of System Simulation, 2016, 28 (12):

2933-2938.)

[12] Li Zhenhai, Luo Zhicai, Wang Haihong, *et al*. Visualization of gravity vector field using improved FLIC algorithm [J]. Geomatics & Information Science of Wuhan University, 2011, 36 (3): 276-270.

[13] Matvienko V, Kruger J. Explicit frequency control for high-quality texture-based flow visualization [C]// Proc of IEEE Scientific Visualization Conference. 2015: 41-48.

[14] Karch G K, Sadlo F, Weiskopf D, *et al*. Visualization of 2D unsteady flow using streamline-based concepts in space-time [J]. Journal of Visualization, 2016, 19 (1): 115-128.

[15] 李伟，宁建国. 着色噪声的 LIC 算法 [J]. 计算机仿真，2003, 20 (3): 30-32. (Li Wei, Ning Jianguo. Coloring-noise line integral covolution algorithm [J]. Computer Simulation, 2003, 20 (3): 30-32.)

[16] Nakatsukasa Y, Soma T, Uschmajew A. Finding a low-rank basis in a matrix subspace [J]. Mathematical Programming, 2017, 162 (1-2): 325-361.

[17] Lawonn K, Krone M, Ertl T, *et al*. Line integral convolution for real-time illustration of molecular surface shape and salient regions [J]. Computer Graphics Forum, 2014, 33 (3): 181-190.

[18] Hamid F N A, Rabiei F, Ismail F. Composite group of explicit Runge-Kutta methods [C]// Proc of International Conference on Mathematical Sciences & Statistics. [S. l.] : AIP Publishing LLC, 2016: 577-593.

[19] Hlawatsch M, Sadlo F, Weiskopf D. Hierarchical line integration [J]. IEEE Trans on Visualization & Computer Graphics, 2011, 17 (8): 1148.

[20] Schmunk R B. Panoply netCDF, HDF and GRIB data viewer [EB/OL]. (2018-01-10) [2018-02-25]. https://www.giss.nasa.gov/tools/panoply/colorbars/.